Package 'pqr'

January 30, 2020

Type Package

Title pqr

Version 1.0.1

Author Chao Cheng [aut],
Xingdong Feng [aut],
Jian Huang [aut],
Xu Liu [aut,cre]

Maintainer Xu Liu and Chao Cheng <liu.xu@shufe.edu>

Description A regularized projection score method is proposed for estimating treatment effects in quantile regression in the presence of high-dimensional confounding covariates. This method is based on an estimated projection score function of the low-dimensional treatment parameters in the presence of high-dimensional confounding covariates. We propose one-step algorithm and a reffitted wild bootstrapping approach for variance estimation. This enables us to construct confidence intervals for the treatment effects in the high-dimensional circumstances.

License GPL-2

Depends R (>= 3.1.0), SparseM, quantreg

Imports Rcpp (>= 0.11.15), RcppEigen (>= 0.3.2.3.0)

LinkingTo Rcpp, RcppEigen

RoxygenNote 6.0.1

NeedsCompilation yes

Repository github

URL https://github.com/xliusufe/pqr

Encoding UTF-8

R topics documented:

pqr-package *Regularized projection score estimation of treatment effects in high-dimensional quantile regression*

Description

A regularized projection score method is proposed for estimating treatment effects in quantile regression in the presence of high-dimensional confounding covariates. This method is based on an estimated projection score function of the low-dimensional treatment parameters in the presence of high-dimensional confounding covariates. We propose one-step algorithm and a reffitted wild bootstrapping approach for variance estimation. This enables us to construct confidence intervals for the treatment effects in the high-dimensional circumstances.

Author(s)

Xu Liu

Maintainer: Xu Liu <liu.xu@sufe.edu.cn>

References

Cheng, C., Feng, X., Huang, J. and Liu, X. (2020). Regularized projection score estimation of treatment effects in high-dimensional quantile regression. Manuscript.

inferen *Provide CI of individual coefficient of high-dimensional quantile regression*

Description

This function provides the confidence intevals of individual coefficient of high-dimensional quantile regression by a regularized projection score method for estimating treatment effects. One-step estimation procedure can speed up computation, and the Bootstrap method can narrow the length of CI.

Usage

```
inferen(y,x,z,tau,method="OneStep",pen="glasso",eps=1e-6,
        sim.level=0.85,iter.num=100,RCV=F,K=1,weights=NULL,B=1000)
```

Arguments

y	The response, a vector of size n
x	The treatment effects, a matrix with dimension $n \times p$
z	The confounders a matrix with dimension $n \times q$
tau	The given quantile, a scale in the unit inteval
method	The method including "OneStep", "Iterative". "OneStep" denotes one-step method (Feng et al. 2019); "Iterative" denotes that the iteration stops when algorithm conveges. Default is "OneStep".

pen	The penalty including "glasso" and "lasso". "glasso" denotes the grouped lasso that is used in the regression of treatment effect on confounders; "lasso" denotes the lasso. Default is "glasso".
eps	The perturbation when the proposed algorithm is used. Default is epsilon=1e-6.
sim.level	The length of tuning parameter α which is selected automatically. Default is 50.
iter.num	The number of iteration if method="Iterative" is used. Default is 100.
RCV	Use refitted cross validation method and wild bootstrap to estimate the asymptotic covariance matrix. Default is False.
K	The number of repeated RCV. Default is 1.
weights	The weights used for wild bootstrap; if not specified (=NULL). Default is NULL.
B	The size for bootstrap. Default is 1000.

Value

ests	Estimator of β. It is a list.
covs	Covariance matrix of β. It is a $d \times d$-matrix.

Author(s)

Chao Cheng, Xingdong Feng, Jian Huang and Xu Liu (liu.xu@sufe.edu.cn)

References

Cheng, C., Feng, X., Huang, J. and Liu, X. (2020). Regularized projection score estimation of treatment effects in high-dimensional quantile regression. Manuscript.

Examples

```
library(pqr)

n <- 50
d <- 3
s <- 3
p <- 20
alpha <- 0.95
beta <- rep(3,d)
eta <- c(rep(3,s),numeric(p-s))
x <- matrix(rnorm(n*d),n,d)
z <- matrix(rnorm(n*(p-1)),n,p-1)
y <- x%*%beta + cbind(1,z)%*%eta + rnorm(n)
fit <- inferen(y,x,z,tau=0.5)
ests <- fit$ests
est.coef <- ests$coef
boot.var <- diag(fit$cov)
lbounds <- ests$coef - qnorm((1+alpha)/2)*sqrt(boot.var)
ubounds <- ests$coef + qnorm((1+alpha)/2)*sqrt(boot.var)
counts <- ifelse(lbounds<beta&beta<ubounds,1,0)
```

mvr *Estimate coefficients of high-dimensional multivariate regression for the column-wise*

Description

This function provides the coefficient matrix estimator of high-dimensional multivariate regression (MVR) with penalty lasso (mcp or scad). The tuning parameter is selected by BIC (the default), AIC, EBIC, CV, or GCV.

Usage

```
mvr(Y, X, Z=NULL, method="BIC", ncv=10, penalty="LASSO", isPenColumn=TRUE,
          lambda=NULL, nlam=50, intercept=TRUE, lam_min=1e-4, eps=1e-6,
          maxstep=20, gamma_pen=2, dfmax=NULL, alpha=1)
```

Arguments

Y	The response, a vector of size n or a matrix of size $n \times q$
X	The covariates to be penalized, a matrix with dimension $n \times p$
Z	The covariates without penalization, a matrix with dimension $n \times d$. The default is NULL.
method	The method to be applied to select parameters. Either BIC (the default), AIC, EBIC, CV, or GCV.
ncv	The number of cross-validation folds. Default is 10. If method is not CV, ncv is useless.
penalty	The penalty to be applied to the model. Either "LASSO" (the default), "SCAD", or "MCP".
isPenColumn	A logical value indicating whether the coefficients associating with X_j that affects whole response y is penalized. Default is TRUE. If isPenColumn is TRUE, the coefficients associating with X_j that affects simultaneously whole response y is penalized for each $j \in \{1, \cdots, p\}$. If isPenColumn is FALSE, the coefficients associating with X_j that affects single response y_l is penalized for each $j \in \{1, \cdots, p\}$, where $l \in \{1, \cdots, q\}$.
lambda	A user-specified sequence of lambda values. By default, a sequence of values of length nlam is computed, equally spaced on the log scale.
nlam	The number of lambda values. Default is 50.
intercept	Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE)?
lam_min	The smallest value for lambda, as a fraction of lambda.max. Default is 1e-3.
eps	Convergence threshhold. The algorithm iterates until the relative change in any coefficient is less than eps1. Default is 1e-4.
maxstep	Maximum number of iterations. Default is 20.
gamma_pen	The tuning parameter of the MCP/SCAD penalty (see details).
dfmax	Upper bound for the number of nonzero coefficients. Default is no upper bound. However, for large data sets, computational burden may be heavy for models with a large number of nonzero coefficients.

alpha	Tuning parameter for the Mnet estimator which controls the relative contributions from the LASSO, MCP/SCAD penalty and the ridge, or L2 penalty. alpha=1 is equivalent to LASSO, MCP/SCAD penalty, while alpha=0 would be equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbitrarily small, but not exactly 0.

Value

Bhat	Estimator of Bhat.
rss	Residual sum of squares (RSS).
activeX	The active set of X. It is a p dimensional vector.
lambda	The sequence of regularization parameter values in the path.
selectedID	The index of lambda corresponding to lambda_opt.
lambda_opt	The value of lambda with the minimum BIC value.
bic	BIC value used to select variables.
muhat	Estimator of intercept μ. It is NULL if intercept is FALSE.
Chat	Estimator of coefficients of Z. Chat is NULL if Z is NULL.
Y	Response Y.
X	Design matrix X.

Author(s)

Chao Cheng, Xingdong Feng, Jian Huang and Xu Liu (liu.xu@sufe.edu.cn)

References

Cheng, C., Feng, X., Huang, J. and Liu, X. (2020). Regularized projection score estimation of treatment effects in high-dimensional quantile regression. Manuscript.

Examples

```
library(pqr)

#example 1
n <- 200
q <- 5
s <- 3
p <- 100
B <- matrix(runif(q*s, 2,3), s)
X <- matrix(rnorm(n*p),n,p)
Y <- X[,1:s]%*%B + matrix(rnorm(n*q),n)
fit <- mvr(Y,X)
fit$activeX
fit$Bhat
which(rowSums(fit$Bhat^2)>0)
fit$muhat

#example 2
n <- 200
q <- 5
s <- 3
d <- 3
```

```
p <- 100
B <- matrix(runif(q*s, 2,3), s)
C <- matrix(runif(q*d, 1,2), d)
X <- matrix(rnorm(n*p),n,p)
Z <- matrix(rnorm(n*d),n)
Y <- X[,1:s]%*%B + Z%*%C + matrix(rnorm(n*q),n)
fit <- mvr(Y,X,Z)
fit$activeX
fit$Bhat
which(rowSums(fit$Bhat^2)>0)
fit$Chat
fit$muhat
```

Index